RSAC 2025 Conference

Many Voices.
One Community.

SESSION ID: PDP-R02

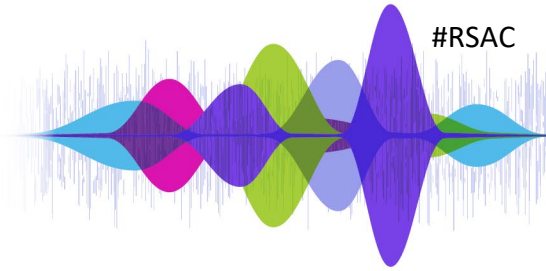
How Should We Protect Keys? Why Key Blocks *Matter*

Richard Kisley

Chief Engineer, IBM HSM
IBM Corporation
https://www.linkedin.com/in/kisley/

Joachim Vance

Chief Security Architect
Verifone
https://www.linkedin.com/in/joachimvance/



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

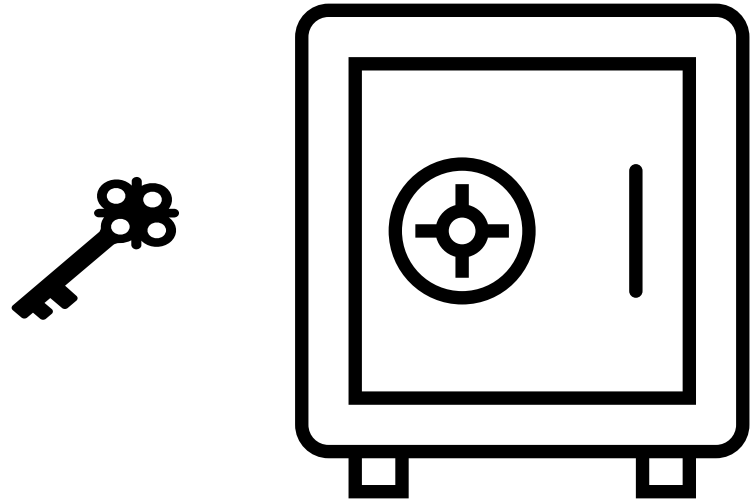
RSAC™ 2025 Conference

What is key protection?

How important are your keys?

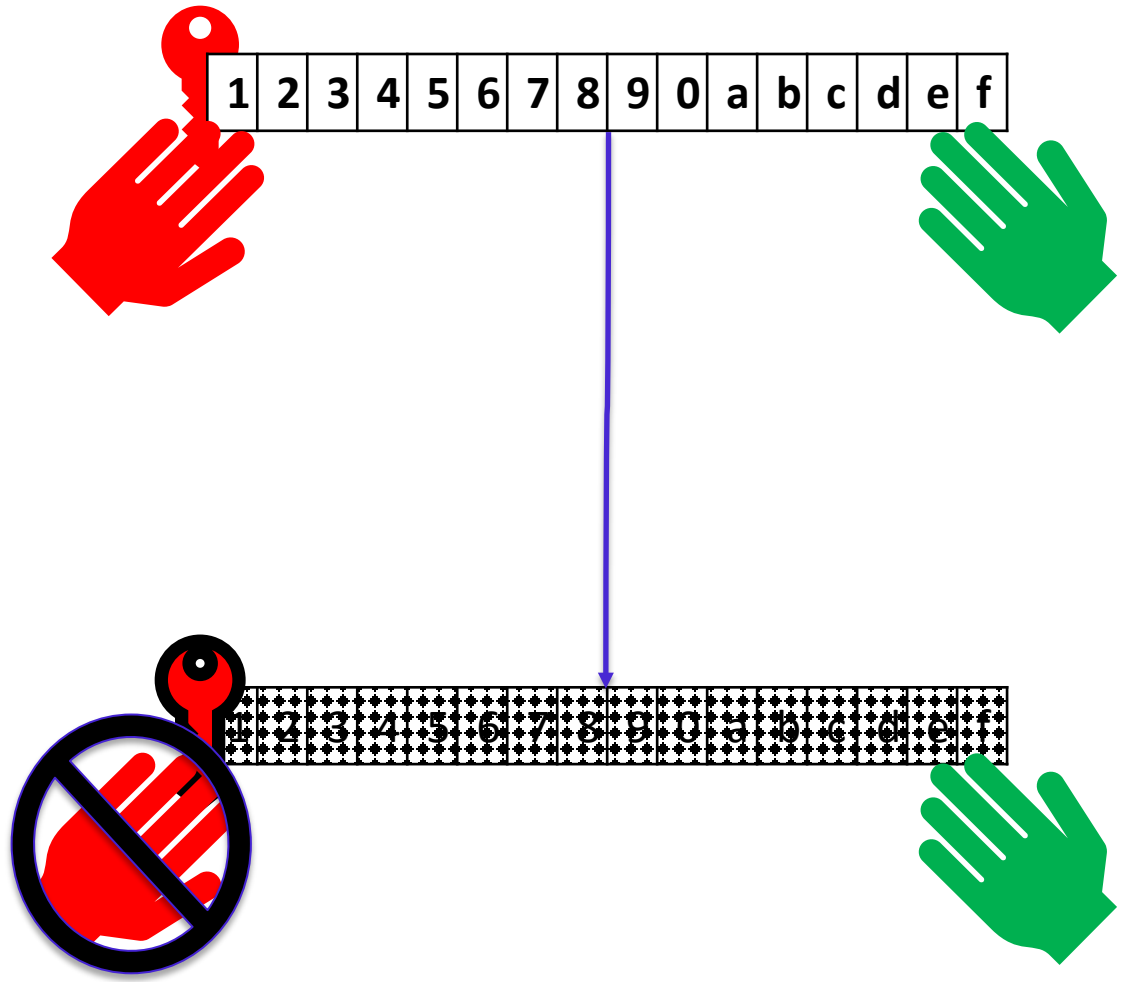
Keys protect your data - how valuable is the data?

- House key
- Browser keys – your payments *and* your medical information
- EMV chip key in your credit card
- …
- Keys between consumer payment network links
- SWIFT keys for cross-border payments
- Nuclear arsenal keys
- *A secure ecosystem transfers the sensitivity of the data to the sensitivity of the key*



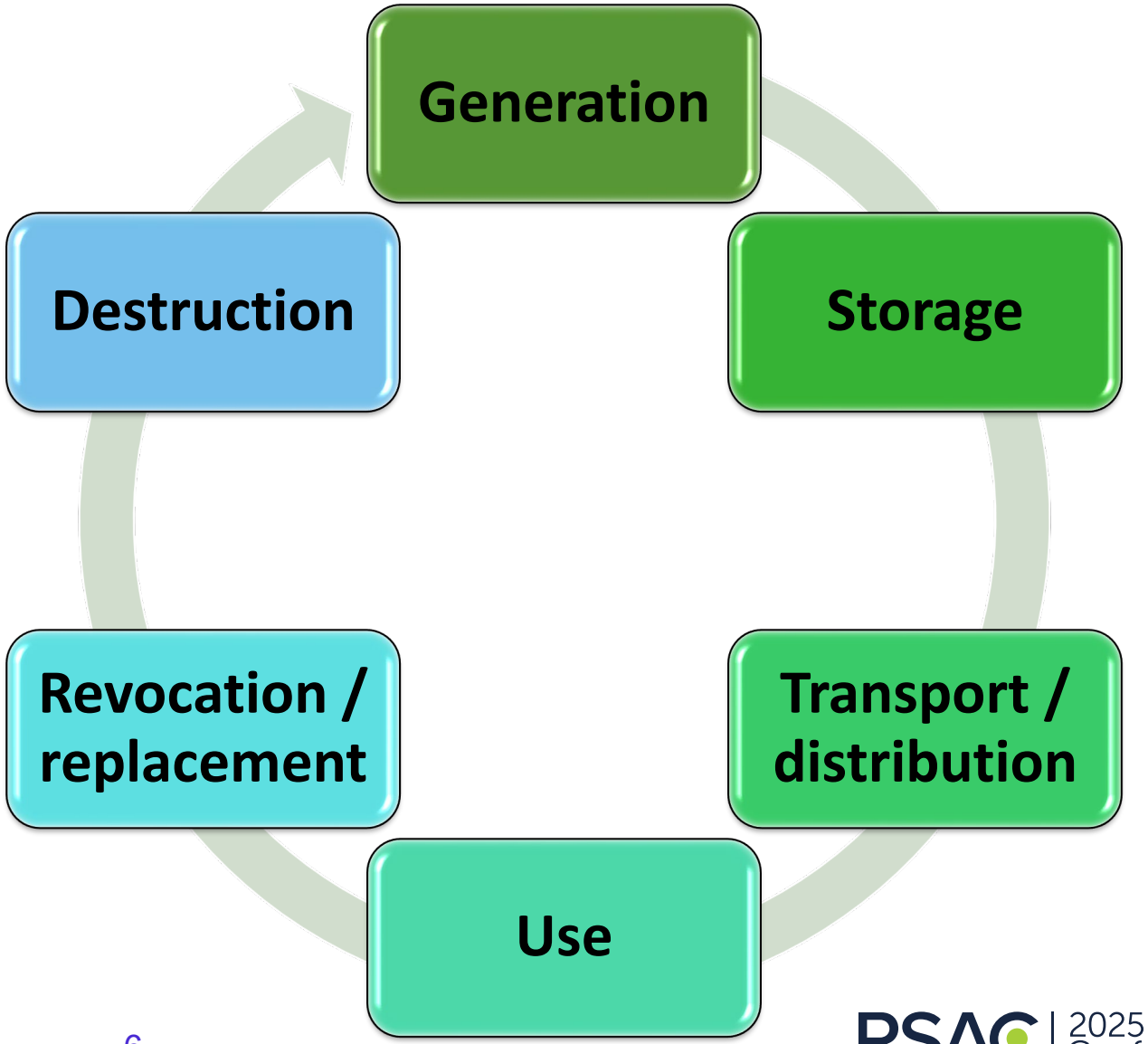
What are you protecting when you protect a key?

- Key value – the ***bits***
- Key controls:
  - ability to ***mis-use*** it
- Create trust:
  - Build ***trusted*** keys
- Key access:
  - ***Who*** can use it



When do you want to protect keys?

- Protect based on the value of the data protected by the key
  - Sometimes you broadcast a key: ‘be sure to drink your ovaltine’
- Consider concern-based protection at all stages of key lifecycle…



RSAC 2025 Conference

Protecting against mis-use

What Can go Wrong?

GOAL	PROBLEM
Protect your data	Breach

What Can go Wrong?

GOAL	PROBLEM
Protect your data	Breach
Protect key value	Key Compromise

What Can go Wrong?

GOAL	PROBLEM
Protect your data	Breach
Protect key value	Key Compromise
Protect key use	Key Misuse Key Reuse

What Can go Wrong?

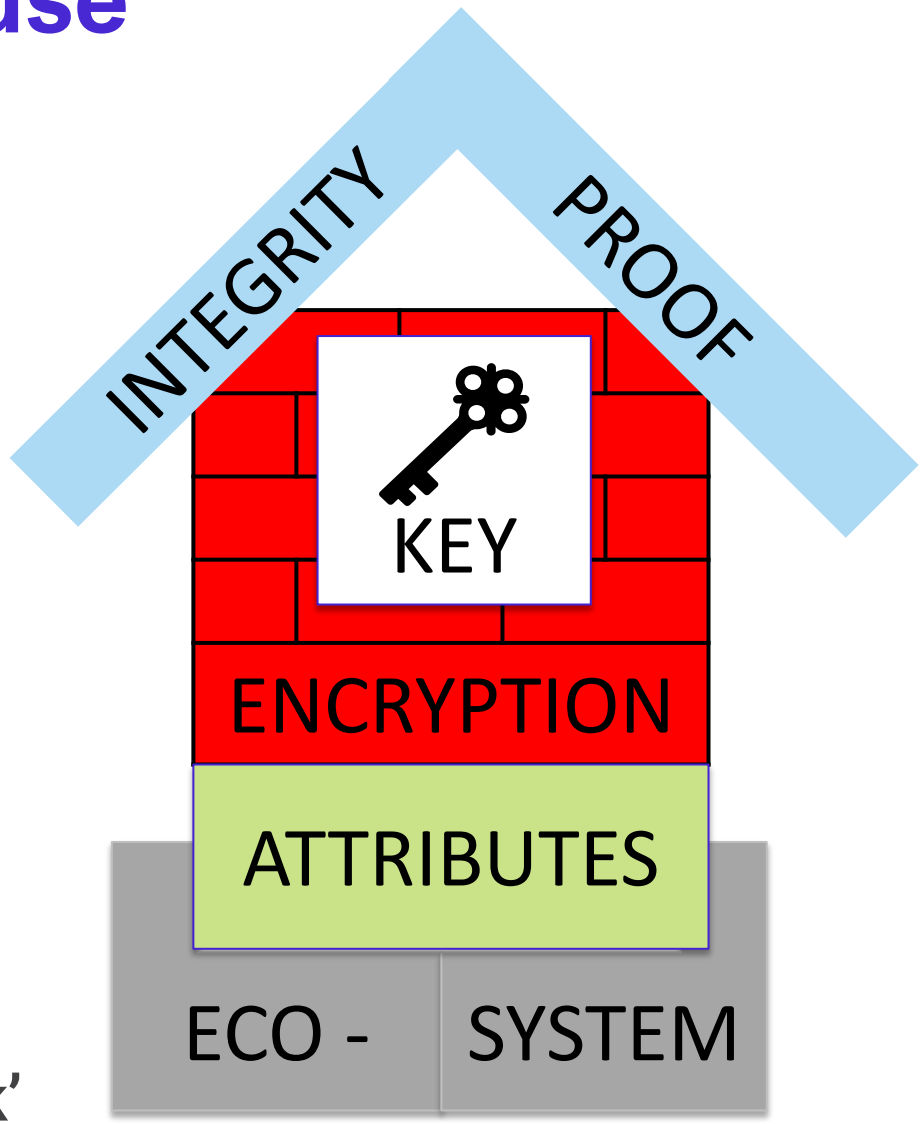
GOAL	PROBLEM
Protect your data	Breach
Protect key value	Key Compromise
Protect key use	Key Misuse Key Reuse
Protect against substitution or change	Key substitution Attribute Substitution

What Can go Wrong?

GOAL	PROBLEM
Protect your data	Breach
Protect key value	Key Compromise
Protect key use	Key Misuse Key Reuse
Protect against substitution or change	Key substitution Attribute Substitution
Maintain control	All of the above

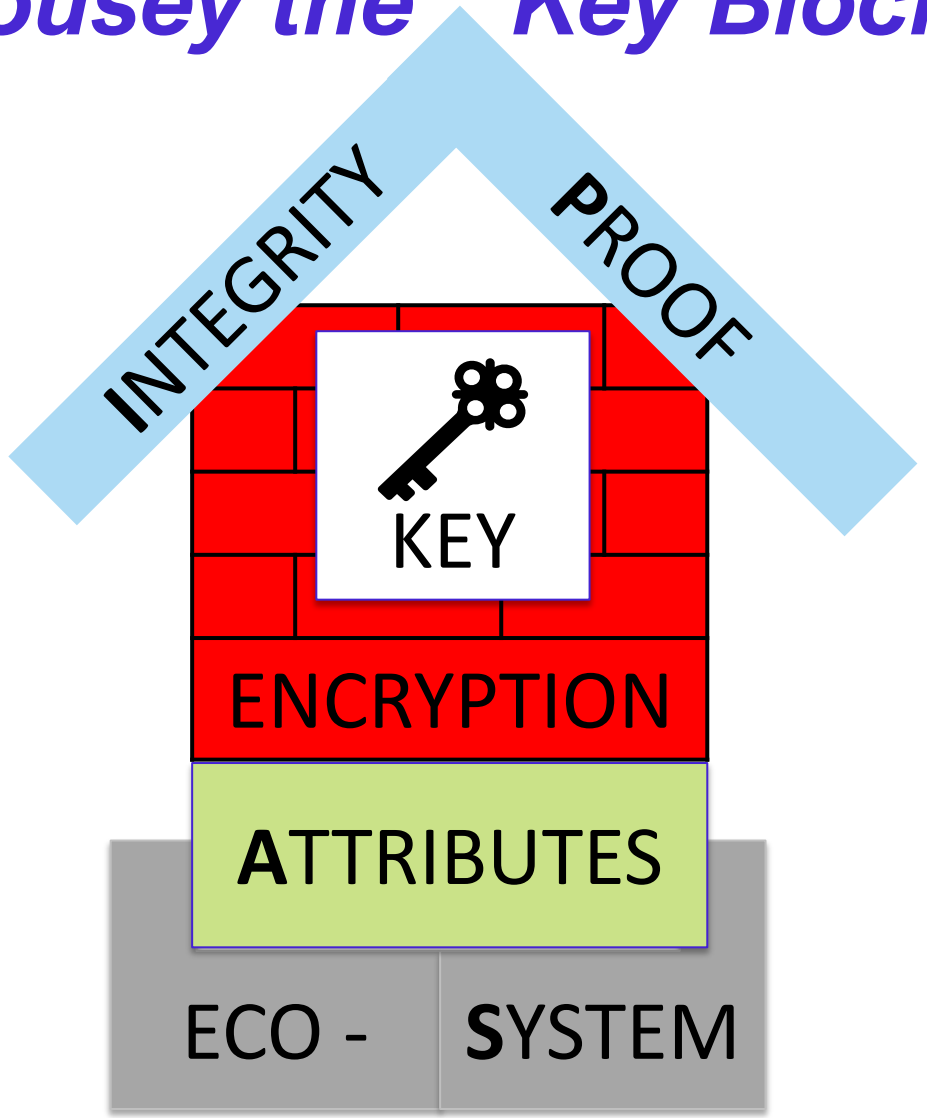
Key protection is building a house

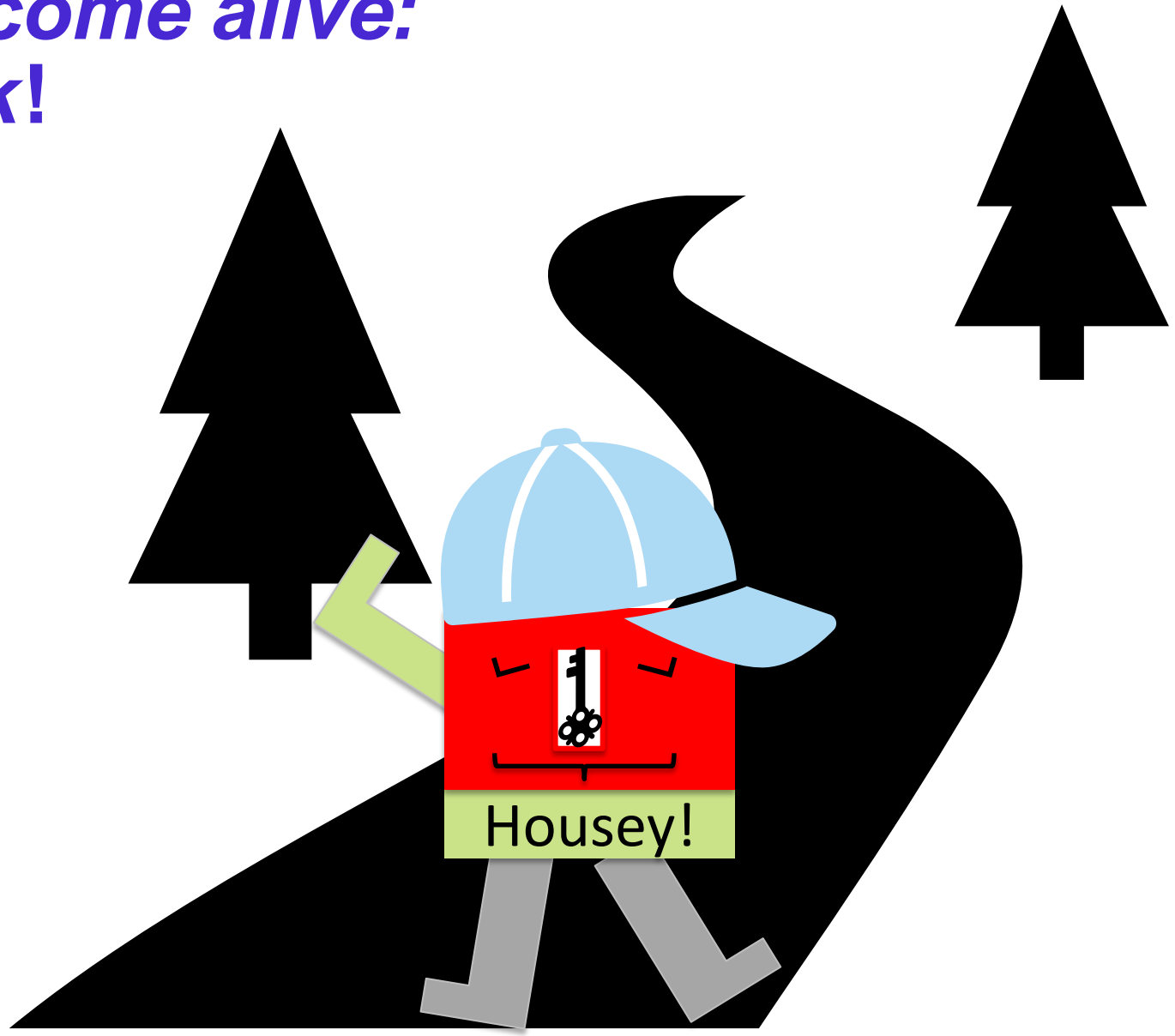
GOAL	FEATURE
Protect your data	Encrypt Data with key
Protect key value	Encrypt the key payload
Protect key use	Attributes
Protect against substitution or change	Integrity & Proof (of authentication)
Maintain control	Secure ecosystem



…this house is called a 'key block'

Let's make key blocks *come alive: Housey the Key Block*!



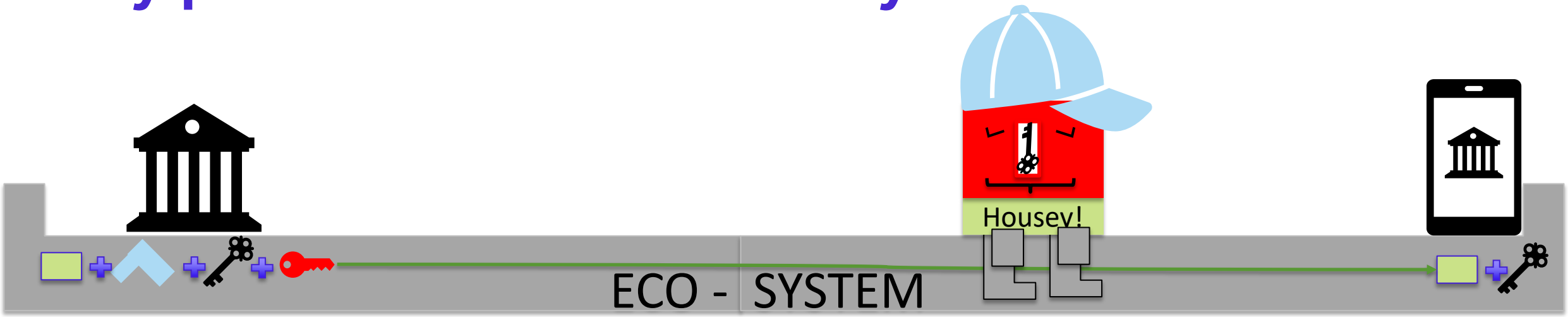


Why is eco-system so important?

The mantra of any good security engineer is: "Security is a not a product, but a process." It's more than designing strong cryptography into a system; it's designing the entire system such that all security measures, including cryptography, work together.

-Bruce Schneier, Blog post: *"Risks of Relying on Cryptography"*

Key protection takes an ecosystem



The ecosystem consists of

- Trusted implementations
- Enforcement of key attributes
- Access controls

What are attributes – why are they important?

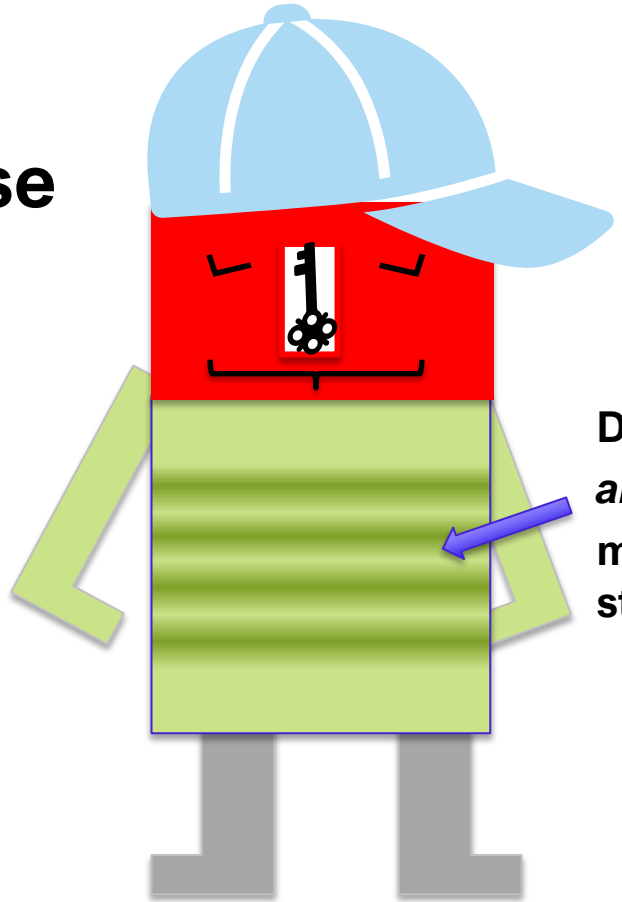
ISO 11568:2023 categories

- Algorithm
- Key Usage
  - encrypt, derive, sign
  - Data operated on
    - ‘Data’, PINs, keys
- Mode
- Exportability

Goal:

1 key =

1 *specific* purpose



Detailed *ab-tributes* make Housey strong!

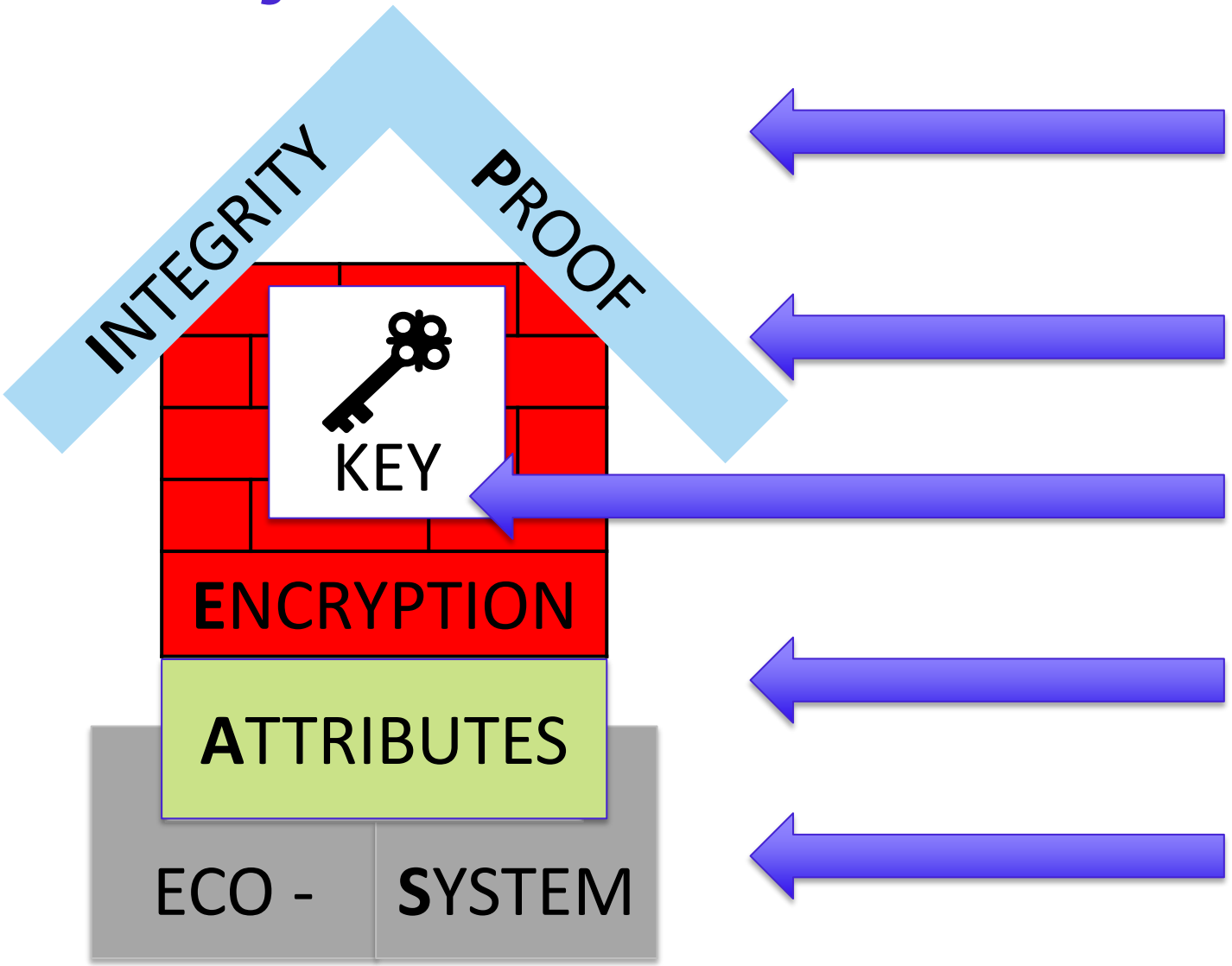
IBM verifone RSAC 2025 Conference

RSAC™ 2025 Conference

Let’s build a key block

Many Voices.
One Community.

What do you start with?



Do you start with the key?

1st: Ecosystem

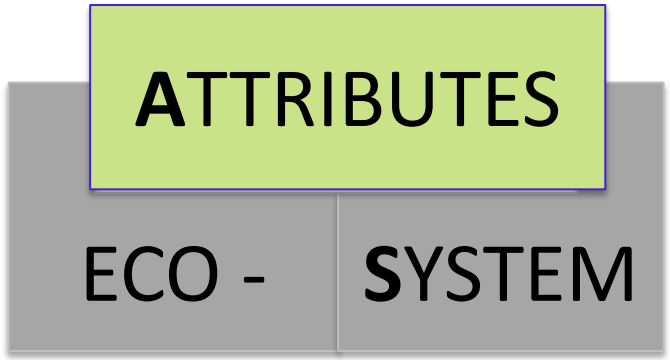
Eco-system questions:

- How is trust established?
  - I am sending a key encrypting key to another company, how can I trust them?
- When you generate keys, where will they be ‘clear’?
- Are key block attributes enforced everywhere?
  - Are you using certified hardware?
- How are access controls managed?

This is the foundation, yet no key block design can answer these, you must build or use a secure eco-system

IBM verifone

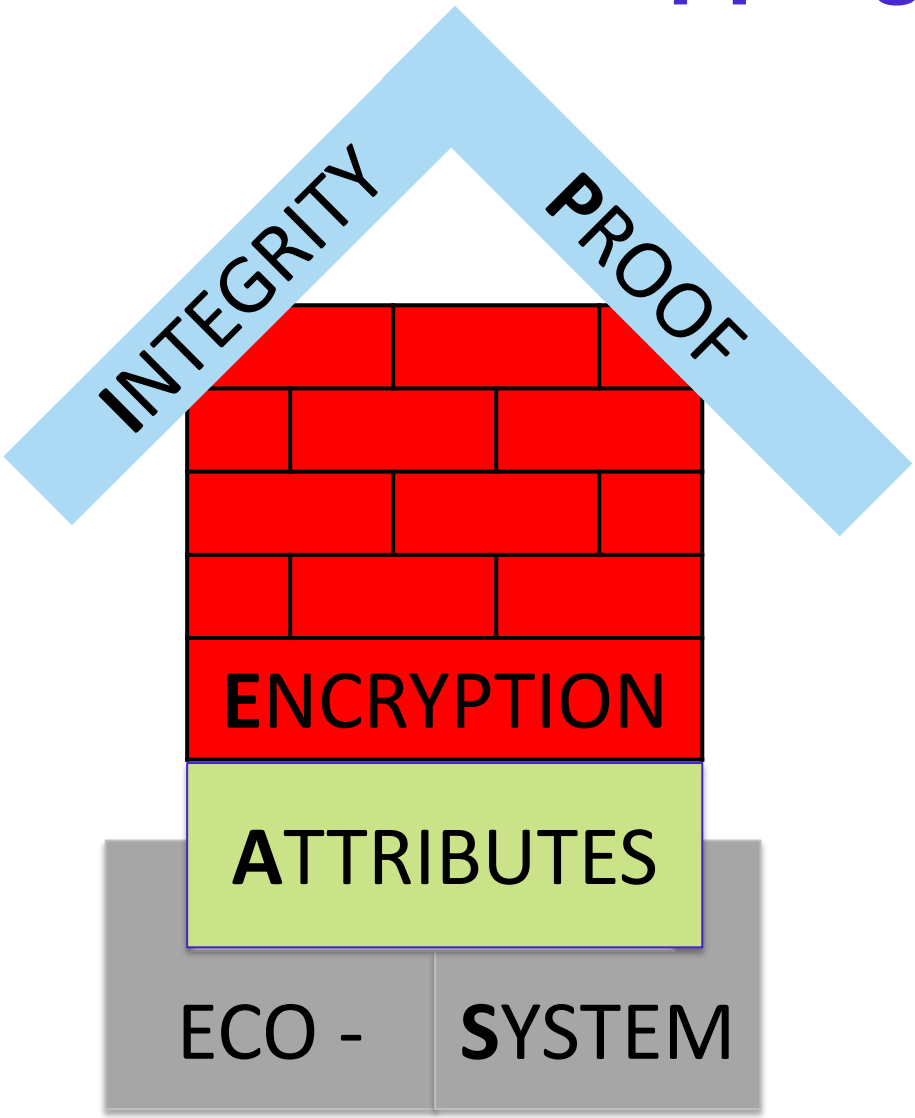
What is next? Attributes



Attribute questions:

- What is the algorithm and mode of use for the key?
- Is the key protecting *regulated* data?
- Can the key be exported from one system to another?
- How do I know which key encrypting key is needed?

What is 3rd? Wrapping mechanism



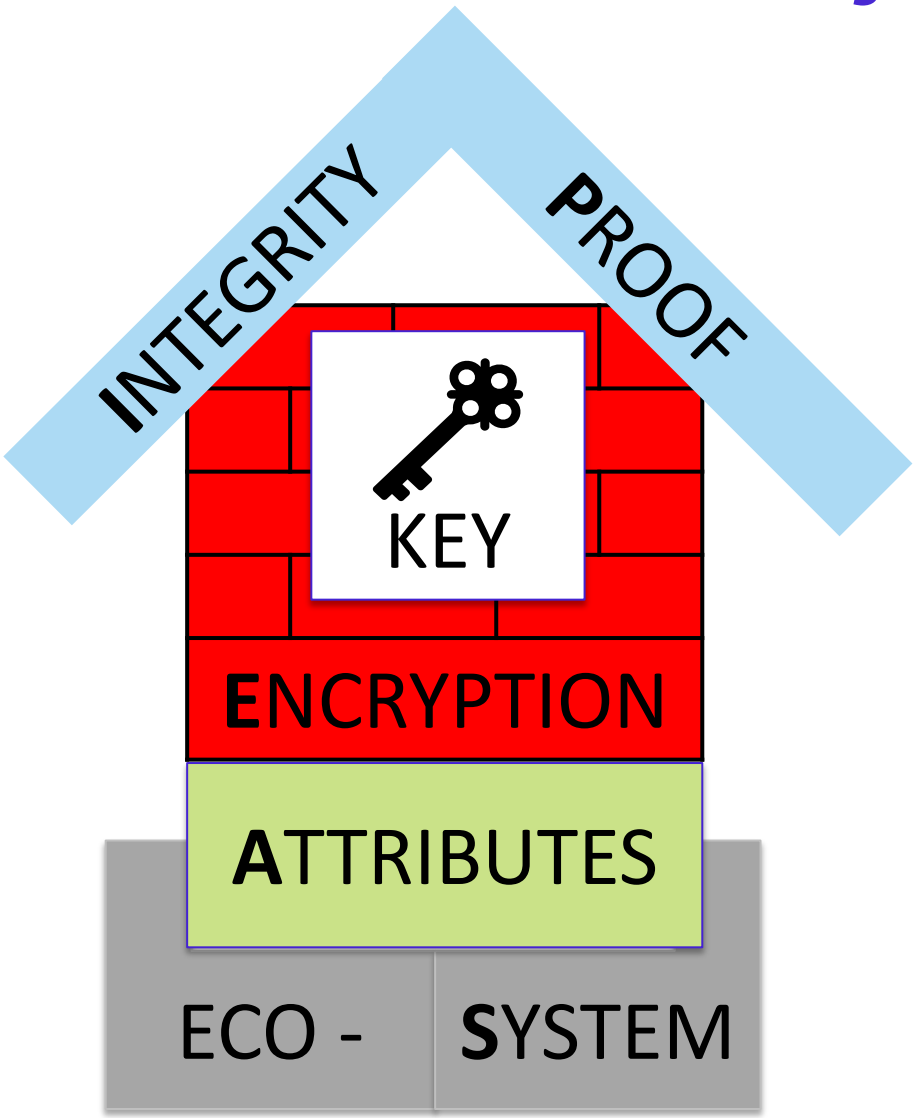
Wrapping =

- Integrity Protection +
- Encryption of the key

Wrapping questions:

- What is the algorithm for the wrapping key?
- What is the integrity protection mechanism?
- Can the key be exported from one system to another?

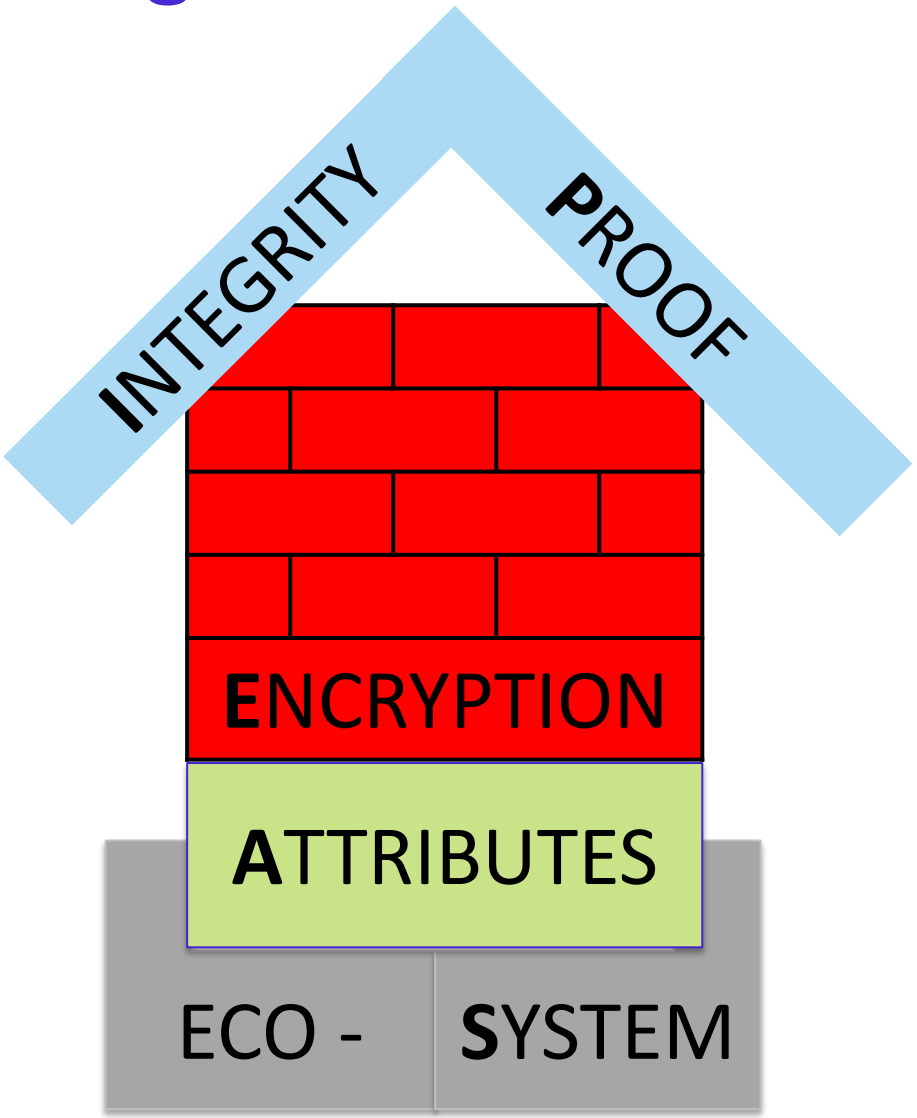
What is last? Put a key in the key block



Key questions:

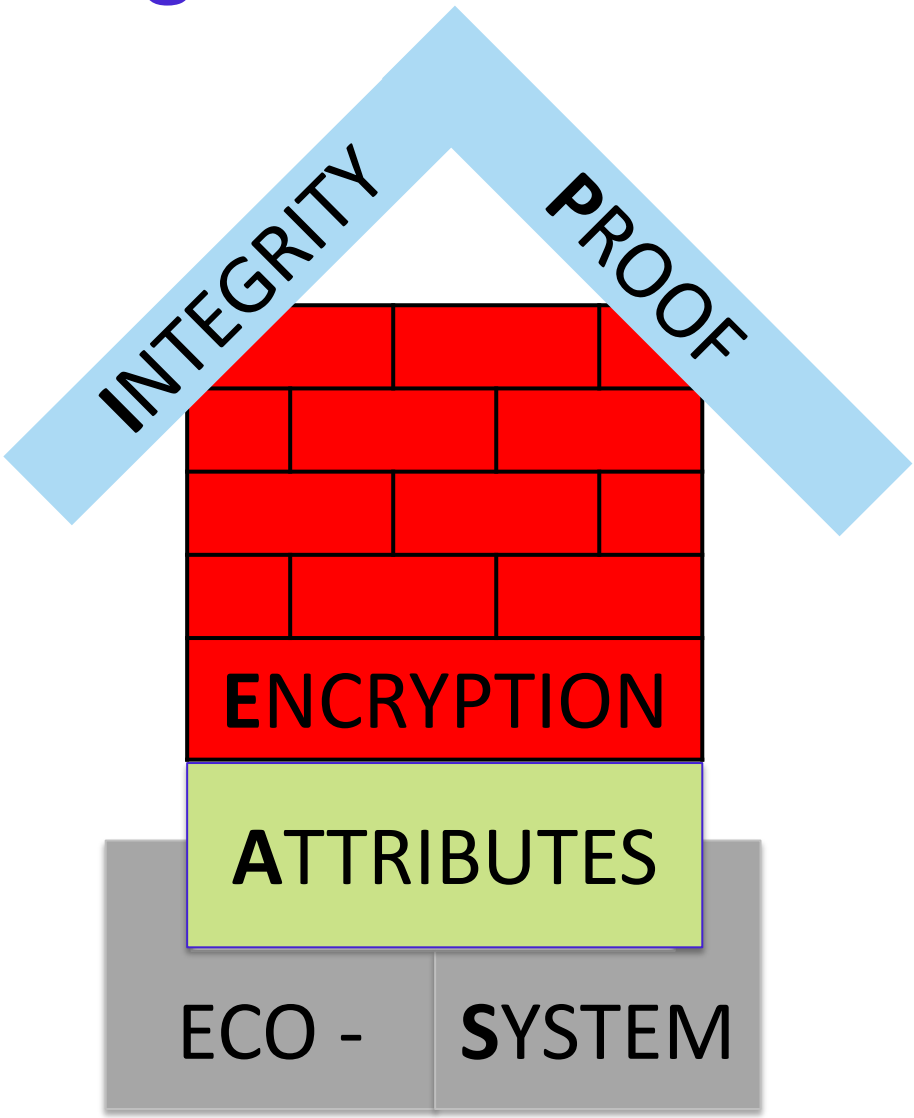
- How is it generated?
- Where is it clear…*in the ecosystem*?

All together - IPEAS



We built a key block, now enjoy the Inner PEAS of mind!

All together - IPEAS



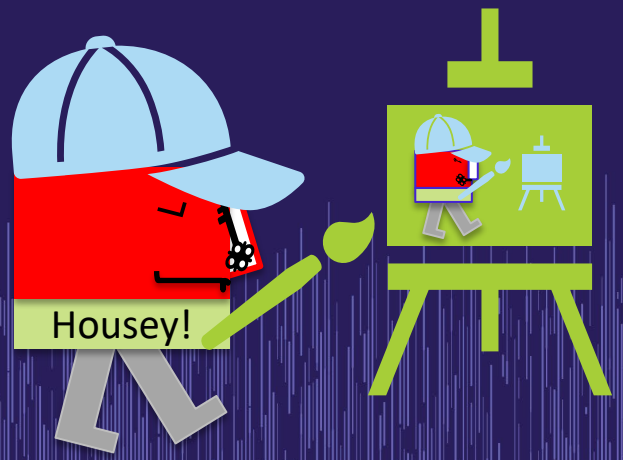
Or if you prefer…have some

I-PE-AS

After building a great key block!

RSAC™ 2025 Conference

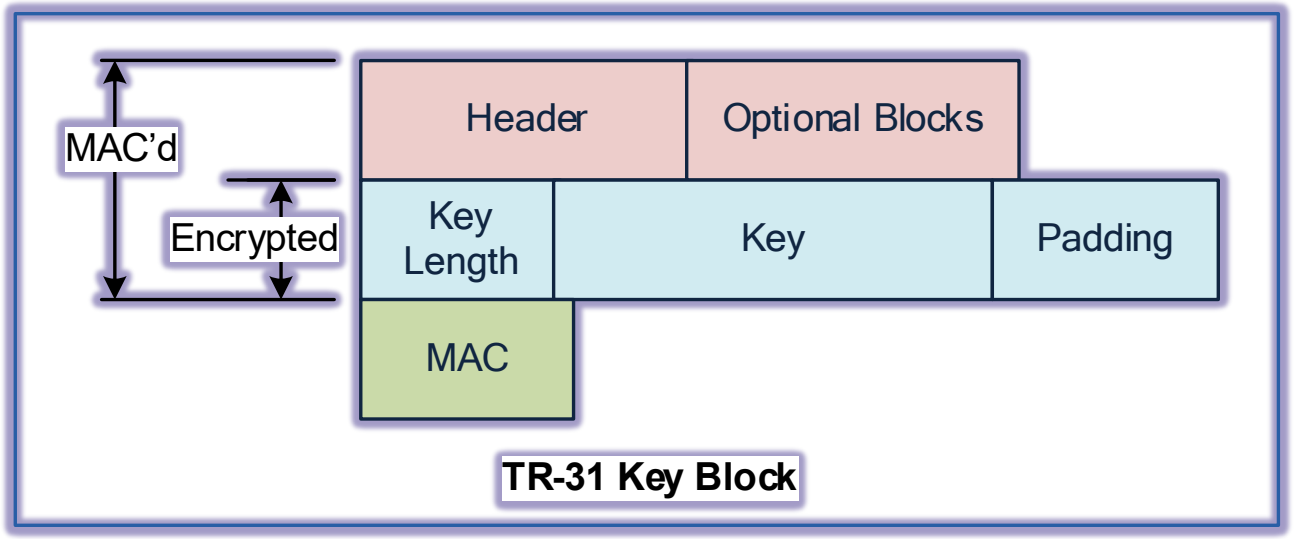
State of the Art

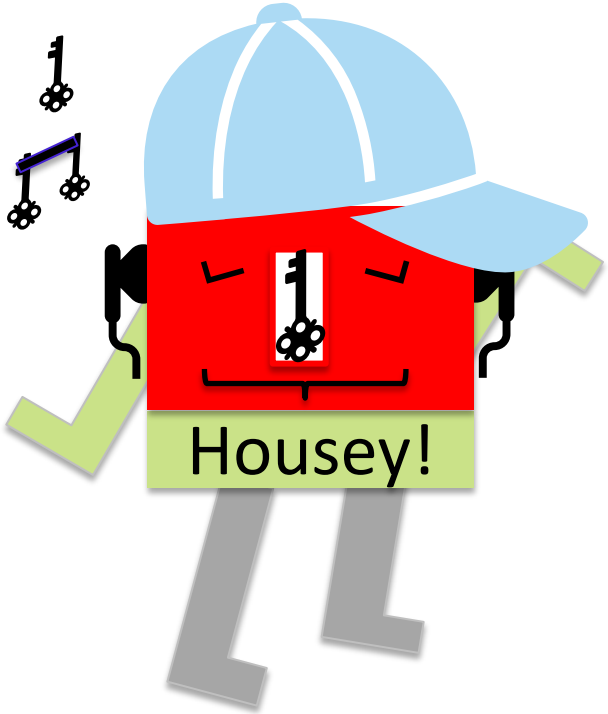


Many Voices.
One Community.

X9.143 (TR-31)

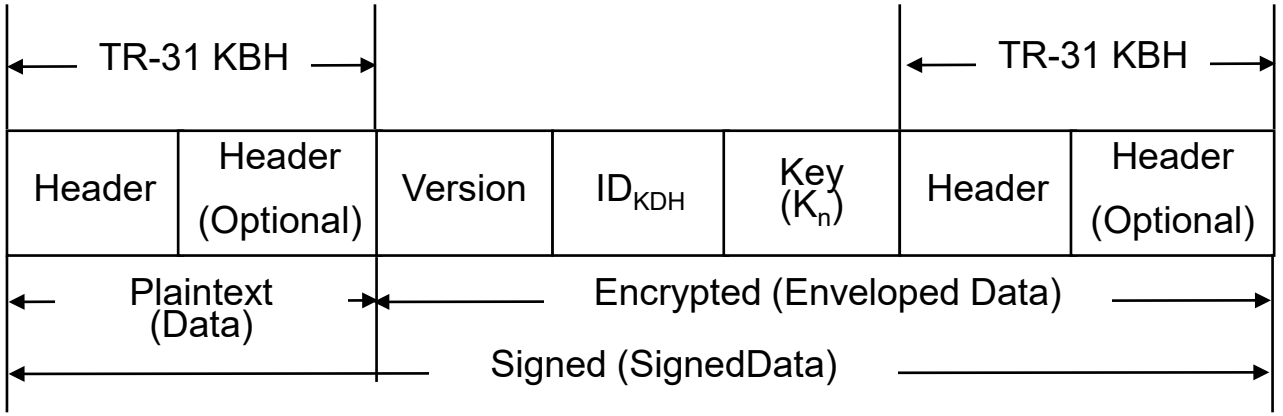
- **Integrity Protection**: MAC
- **Encryption**: CBC
  - 2005: encrypt-then-MAC
  - 2010: MAC-then-encrypt
- **Attributes**
  - map to X9.24 / ISO 11568
- **Eco-System trust**
  - Pre-shared key, hardware…?
- Notes
  - 2 keys derive from ‘wrapping key’
  - 2 methods: 2005 (xor), 2010 (CMAC)

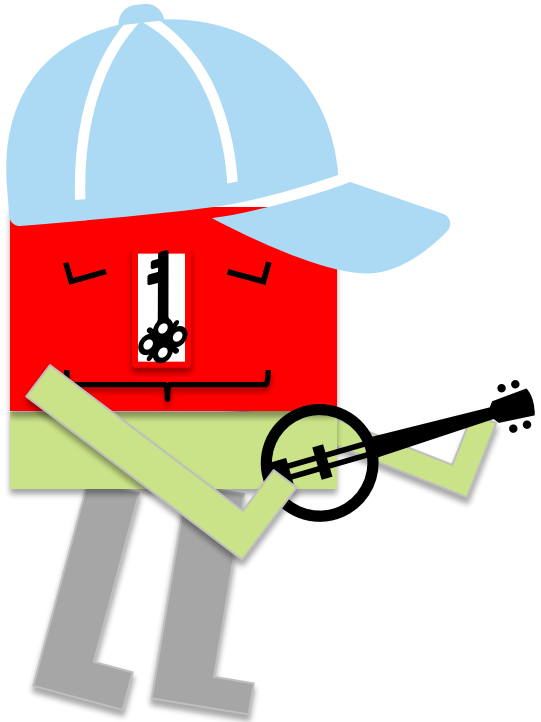




X9 TR-34

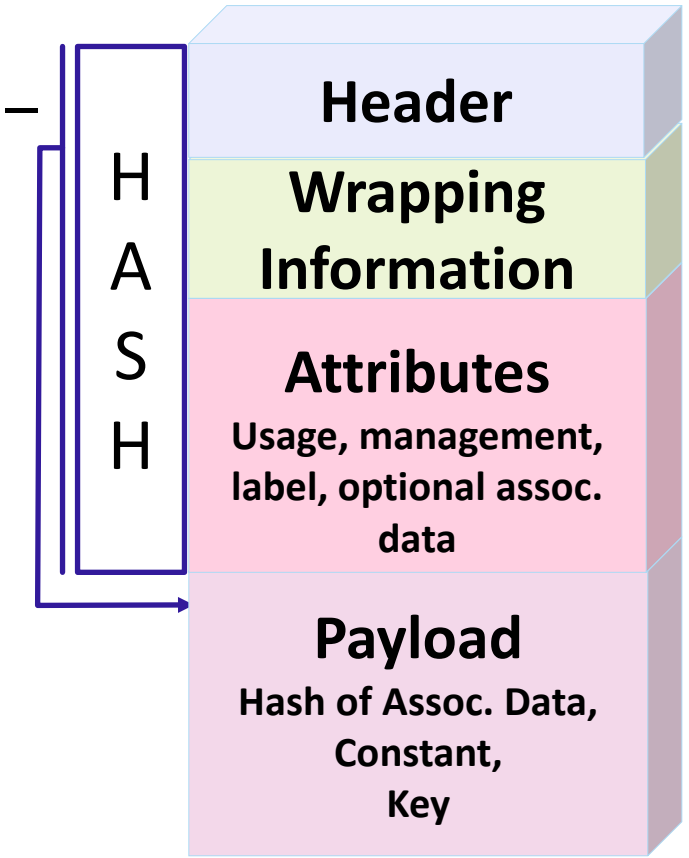
- **Integrity Protection**:
  - RSA signature
- **Encryption**:
  - AES/TDES-CBC with ephemeral symmetric key
  - RSA-OAEP
- **Attributes**
  - TR-31
- **Eco-System trust**
  - Shared CA

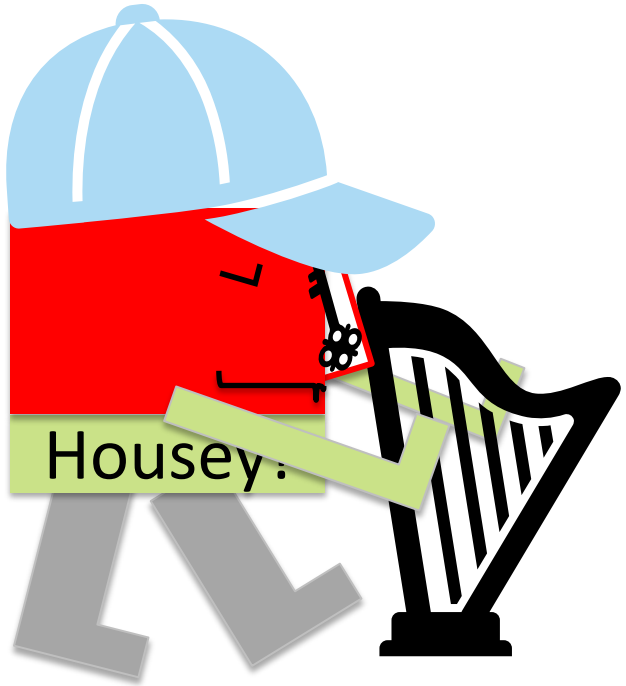




IBM key container – AESKW

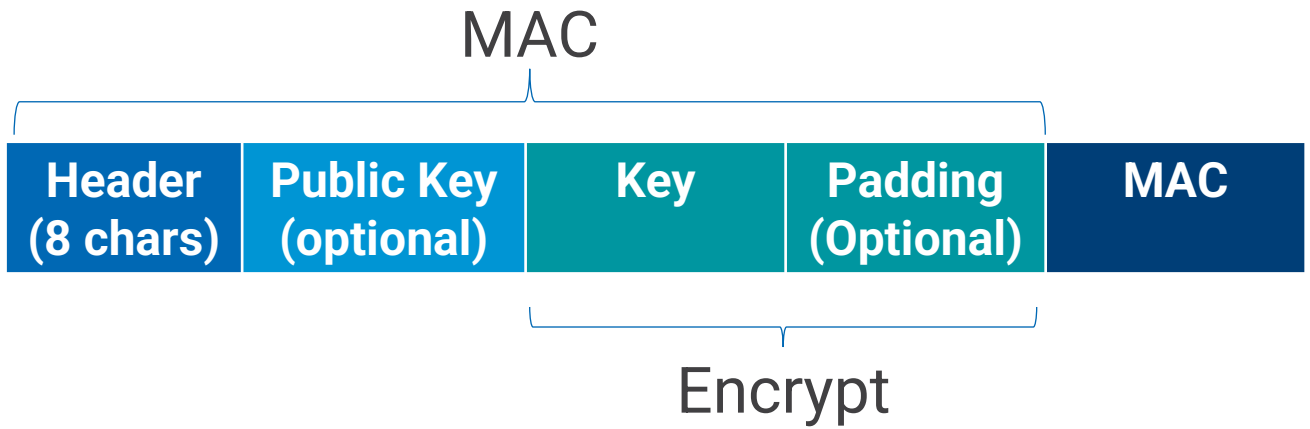
- **Integrity Protection & Encryption**:
  - AESKW (NIST SP800-38F) – authenticated encryption
- **Attributes**
  - More granular than TR-31
- **Eco-System trust**
  - Pre-shared key
- Notes
  - Proprietary, PCI reviewed



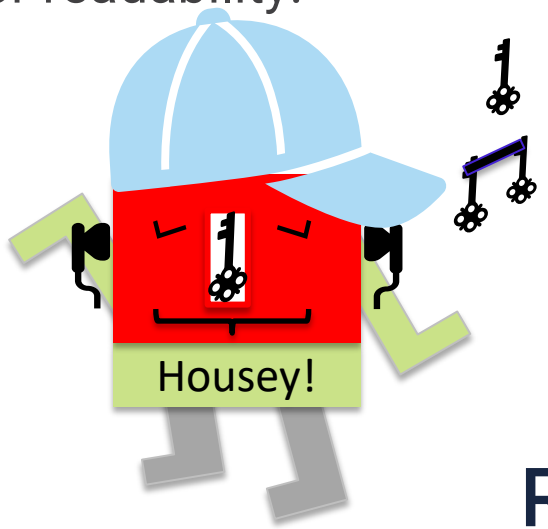


Utimaco Atalla Key Block – AKB

- **Integrity Protection**:
  - TDEA: CBC MAC
  - AES: CMAC
- **Encryption**:
  - TDEA: CBC, encrypt-then-MAC
  - AES: CBC-CS1, MAC-then-encrypt
- **Attributes**
  - More options than X9.143
- **Eco-System trust**
  - Pre-shared key
- **Notes**
  - Proprietary, independent cryptologist (IACR fellow) reviewed
  - Keys derived from MFK via XOR (TDEA) or KDF (AES)

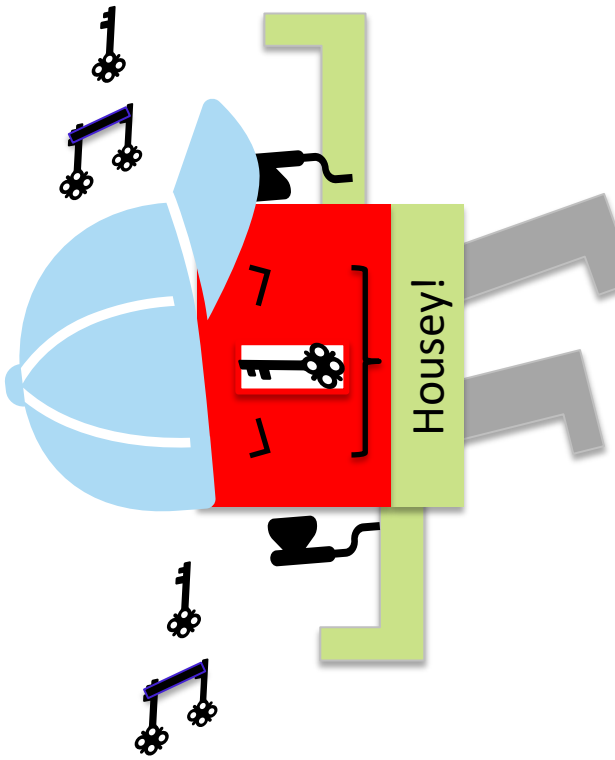


- Human readable header for ease-of-use
- Commas are added after Header and before MAC for readability.



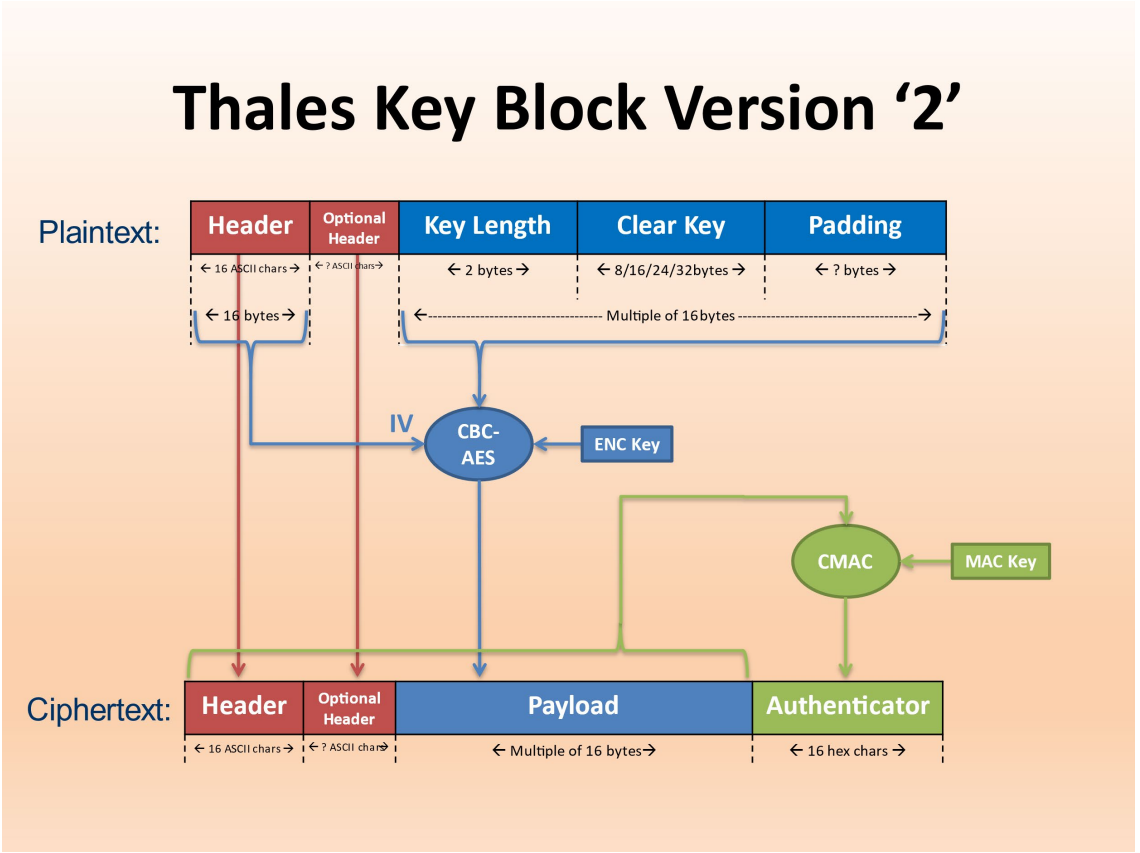
AWS – Payment Cryptography / Cloud HSM

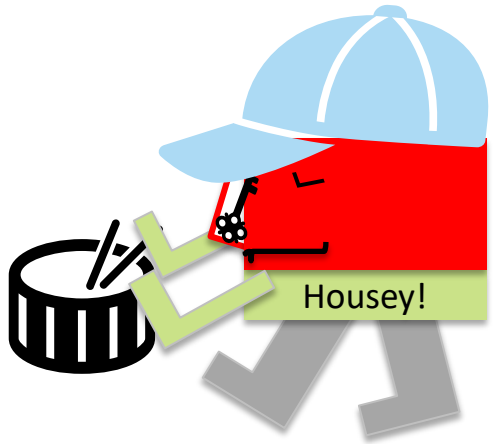
- **Integrity Protection & Encryption**:
  - X9.143/TR-31
  - X9 TR-34
- **Attributes**
  - Derived from X9.143
- **Eco-System trust**
  - Pre-Shared Key
- Notes
  - Uses standards-based key blocks



Thales key block – V2

- **Integrity Protection & Encryption**:
  - Encryption: CBC-AES
  - Authentication: AES-CMAC (encrypt-then-MAC)
- **Attributes**
  - More options than X9.143
- **Eco-System trust**
  - Pre-Shared Key
- Notes
  - Proprietary





RSAC 2025 Conference

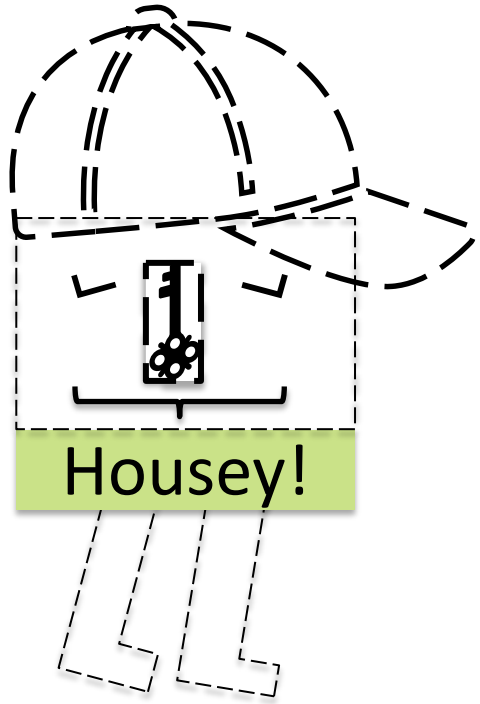
Some popular methods

Many Voices.
One Community.

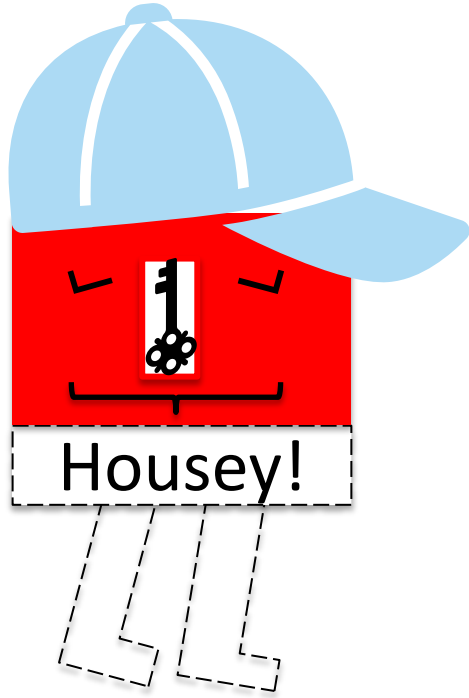
PKCS#11 – encrypted keys

Trust requires a lot of research on implementation and ecosystem

- Standard crypto API from ‘token’
  - ‘token’ = crypto ‘module’ offering the API, defined by the implementor
- Key ‘inside’ the token?
  - **IP**: Check product documentation
    - Could be clear in RAM!
    - Could be stored outside the device
  - **E**: ?
  - **A**: basic key usage, management
- Key outside the token?
  - **IP** & **E**: Many standard mechanisms
  - **A**: **optional**



Inside ‘token’

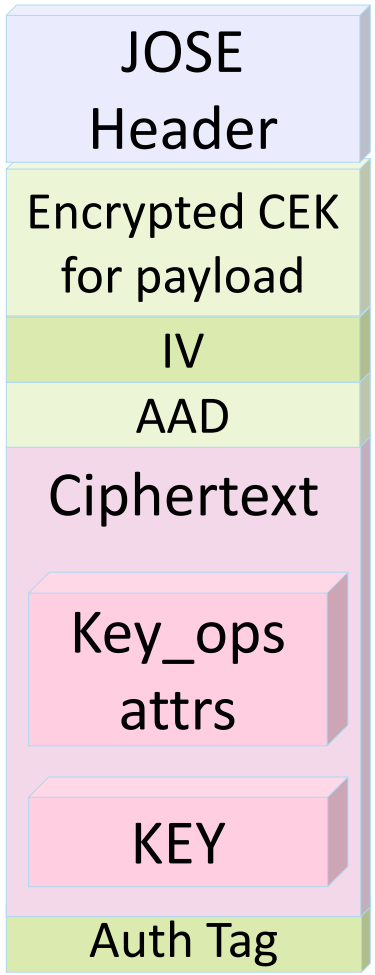


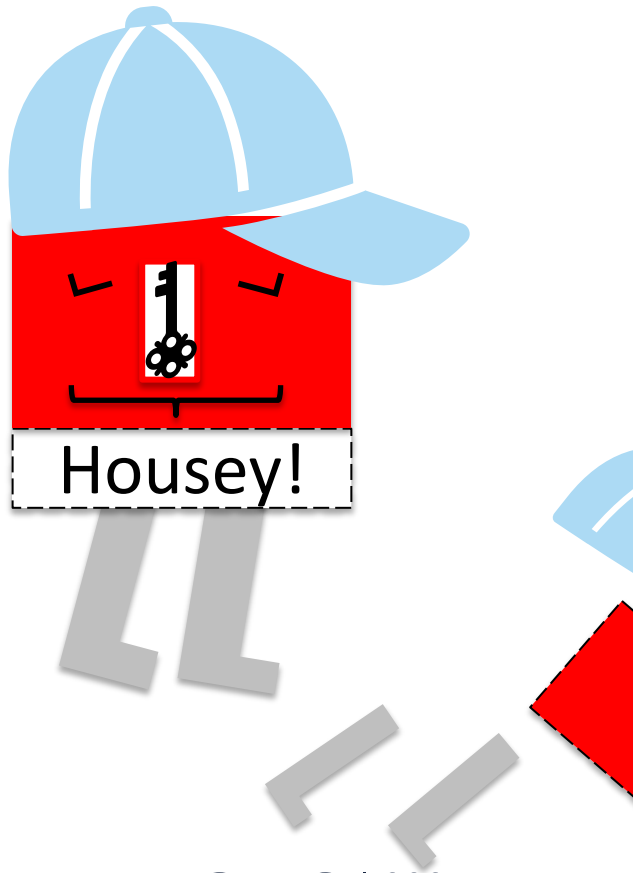
Outside ‘token’

Source: https://docs.oasis-open.org/pkcs11/pkcs11-base/v3.0/

JavaScript Object Notation (JSON) Web Key (JWK)

- Encrypted JWK = JWE with key
  - 1-use content encryption key
  - AAD: Additional Auth Data
- **Integrity Protection & Encryption**:
  - AEAD : MAC + enc, AES-GCM
- **Attributes -** optional
  - Basics – like PKCS#11
- **Eco-System trust**
  - Pre-shared key or common CA

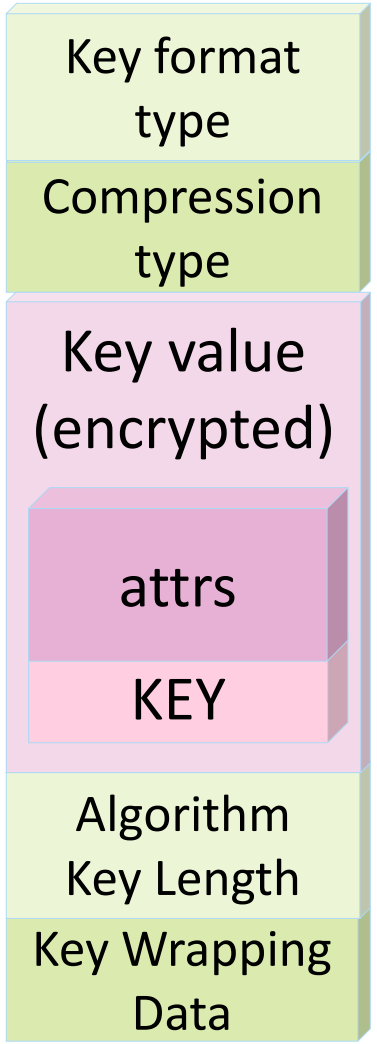


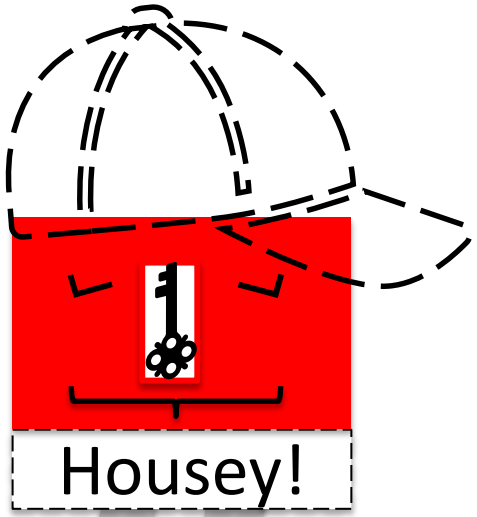


Source: https://datatracker.ietf.org/doc/html/rfc7517

KMIP key container

- **Integrity Protection & Encryption**:
  - Enc with optional MAC/sign
- **Attributes -** optional
  - Basics – like PKCS#11
- **Eco-System trust**
  - Pre-shared key or common CA

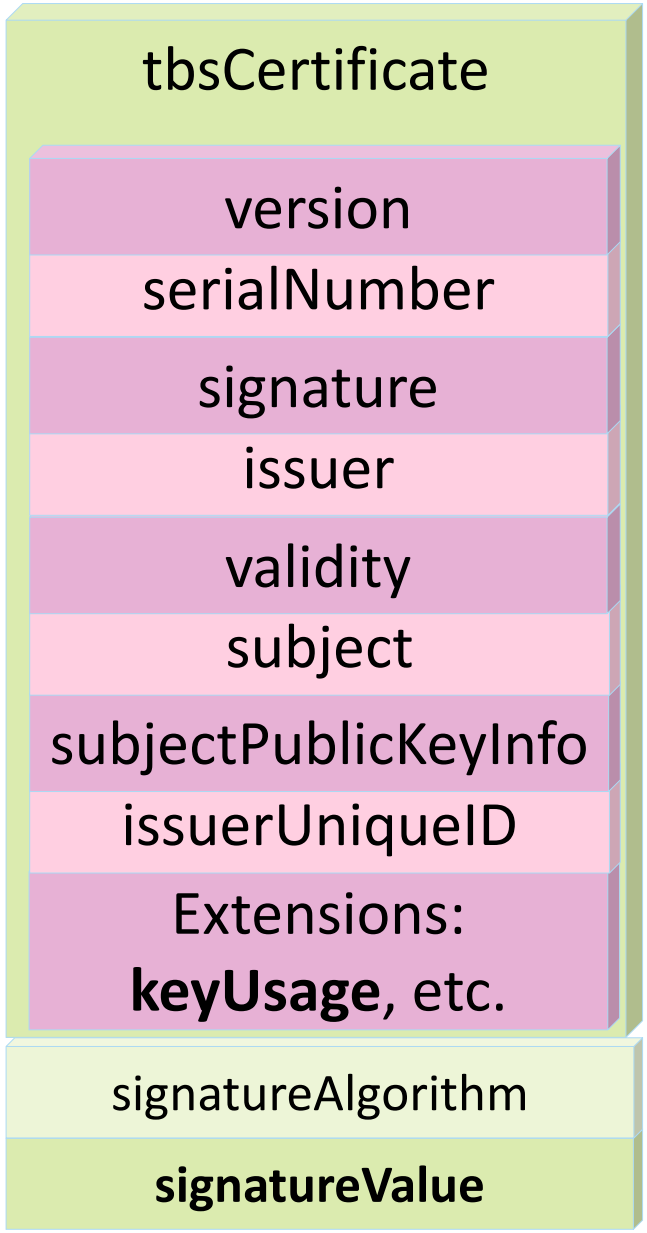


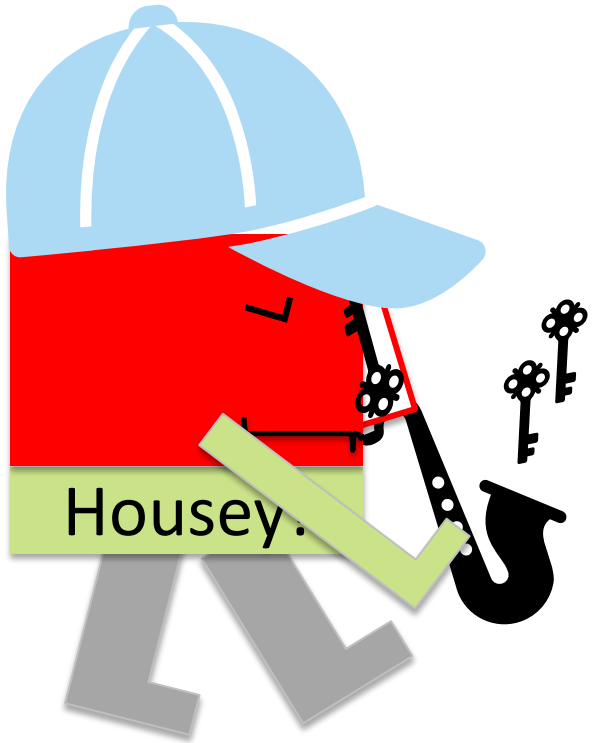


https://docs.oasis-open.org/kmip/kmip-spec/v3.0/kmip-spec-v3.0.html

X.509 Certificate

- **Integrity Protection & ~~Encryption~~**:
  - signature
- **Attributes**
  - keyUsage – required for CA
- **Eco-System trust**
  - Common CA root
- Notes
  - RFC 2459, 3280, 5280 have different validation

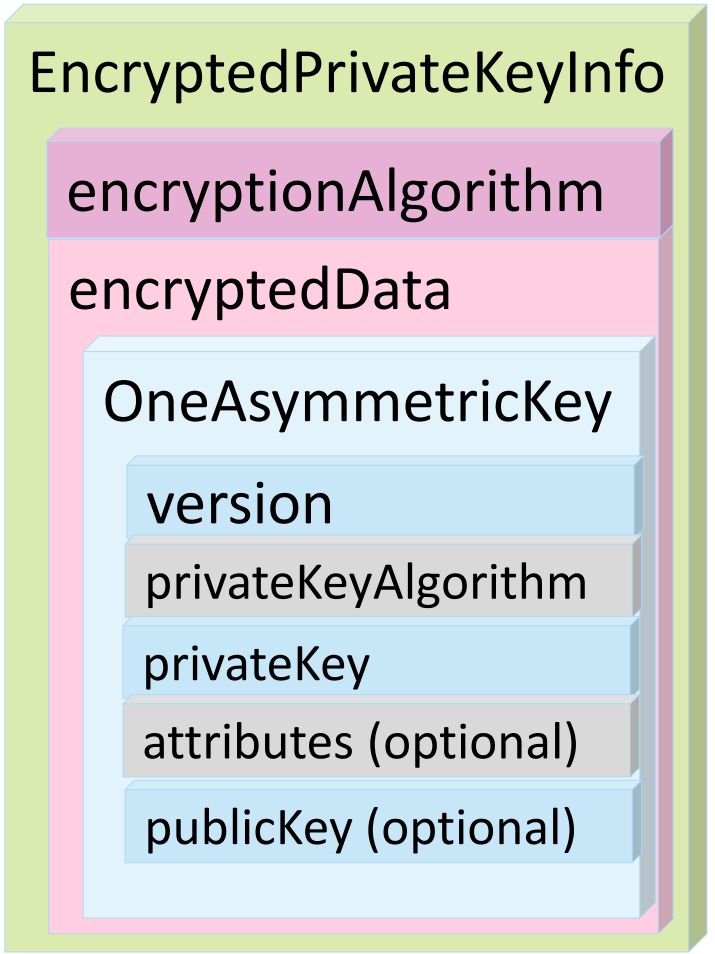


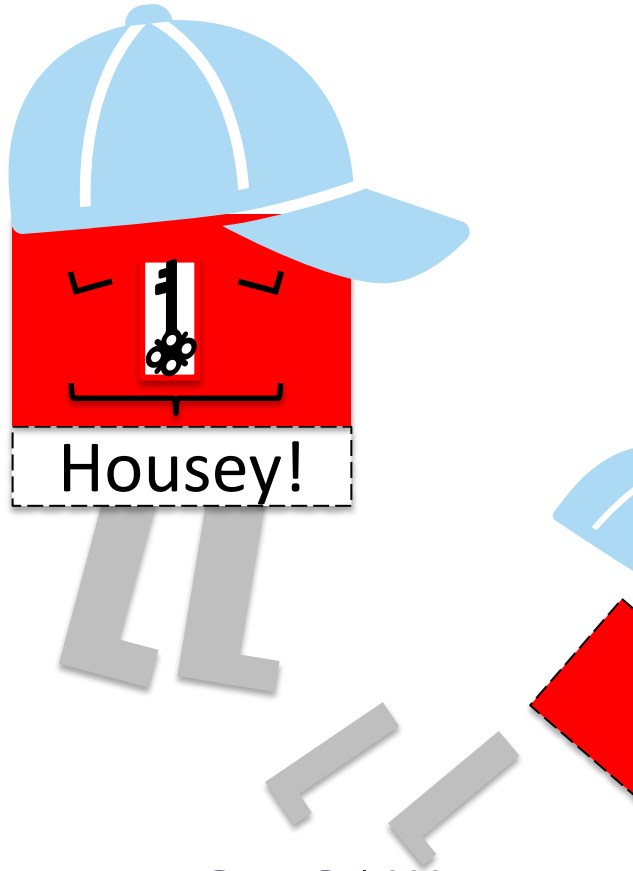


Source: https://datatracker.ietf.org/doc/html/rfc5280#section-4.1

PKCS#8: private key

- **Integrity Protection & Encryption**:
  - Many encryptionAlgorithm options
- **Attributes**
  - For public key, optional
- **Eco-System trust**
  - Pre-shared key or CA
- Notes
  - PKCS#8 used in PKCS#12



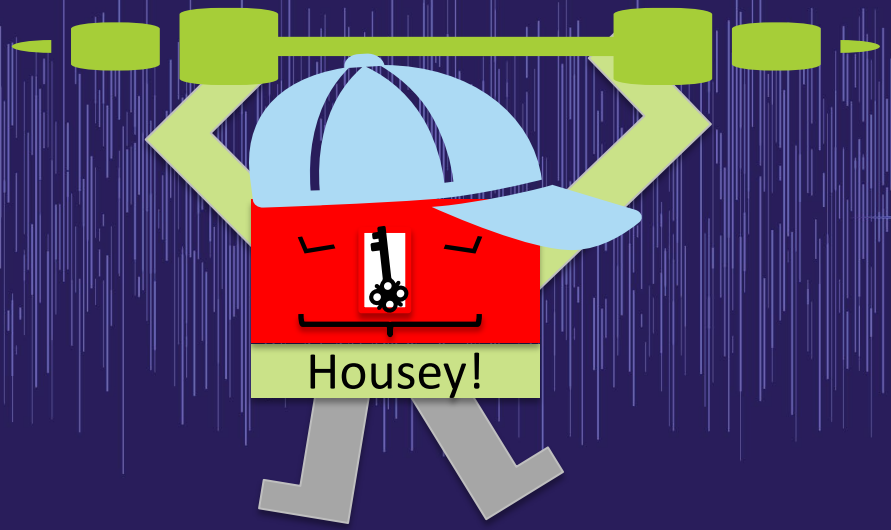


https://datatracker.ietf.org/doc/html/rfc5958

IBM verifone RSAC 2025 Conference

RSAC 2025 Conference

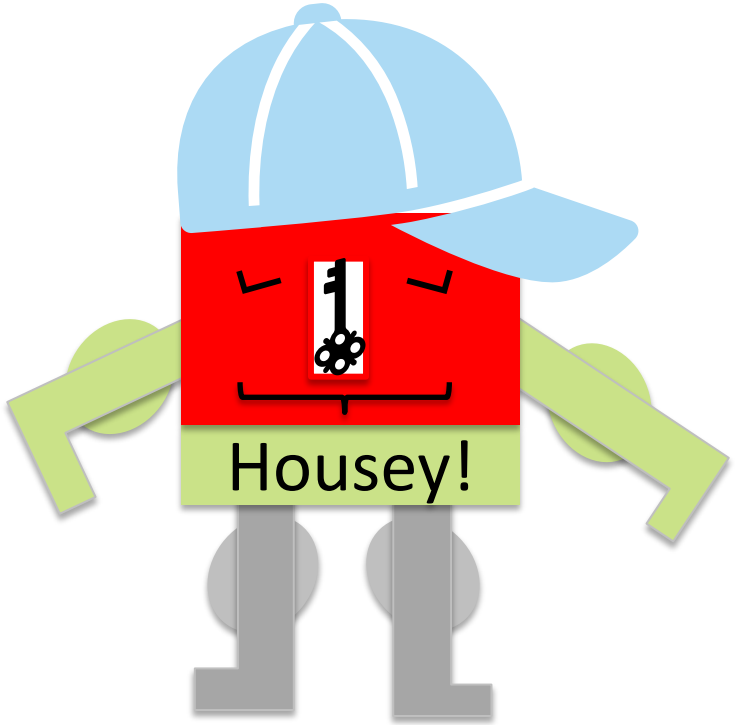
Evolving key protection standards



Many Voices.
One Community.

Guidelines from PCI SSC

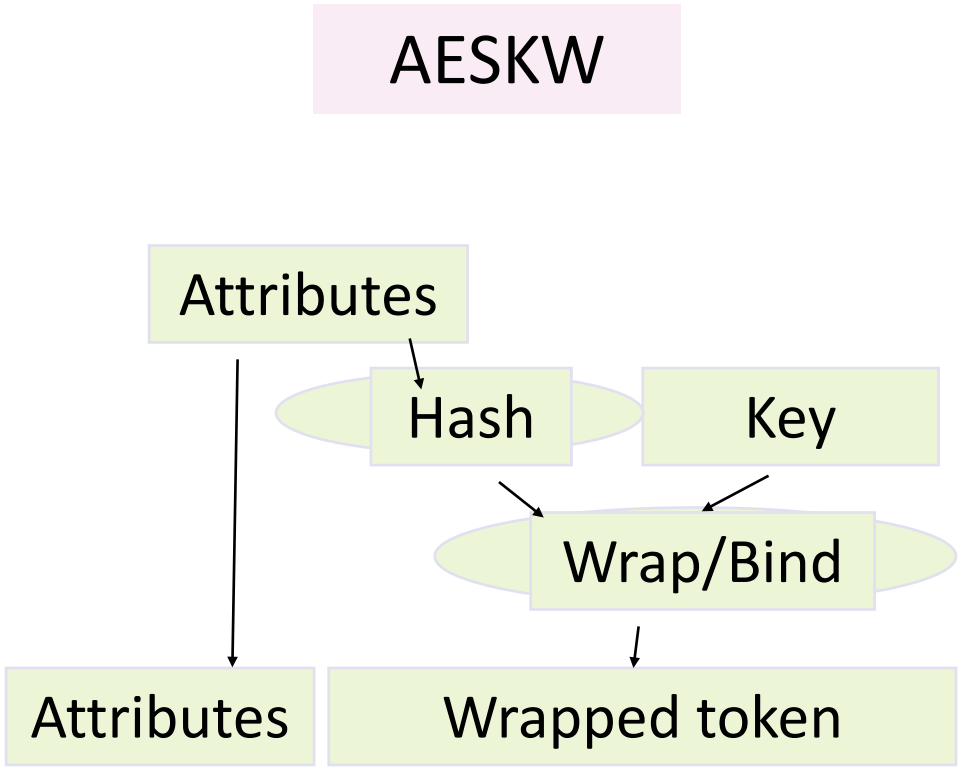
- **Attributes**
  - Operations, algorithm, mode, exportability
- **Key wrap mechanisms**
  - X9.102, X9.143, ISO 19772:mech 2
  - X9 TR-34, RSA OAEP, ECDH->AES
- **Integrity protection**
  - **sym wrap:** part of method, or signature, MAC
  - **asym wrap:** signature, MAC
- **Ecosystem - trust**
  - Pre-shared key, Mutual authentication



https://docs-prv.pcisecuritystandards.org/PIN/Frequently%20Asked%20Questions%20(FAQ)/PCI_PIN_Technical_FAQs_v3_February_2025.pdf

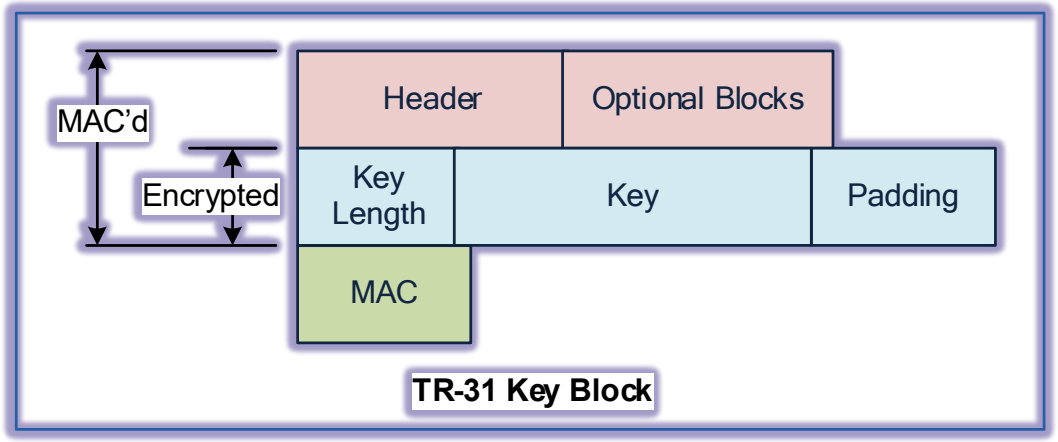
Wrapping methods preferred by NIST

- **NIST SP800-131Ar3-ipd (Oct 2024)**
  - "KW, KWP, CCM, and GCM using AES" approved for keywrap
  - Use of 2 algorithms for encryption & authentication - **deprecated**
- **KW (AESKW)**
  - X9.102, NIST SP800-38F
  - copy or hash of attributes is placed in the payload (AESKW)
  - Encryption rounds with 128-bit cipher



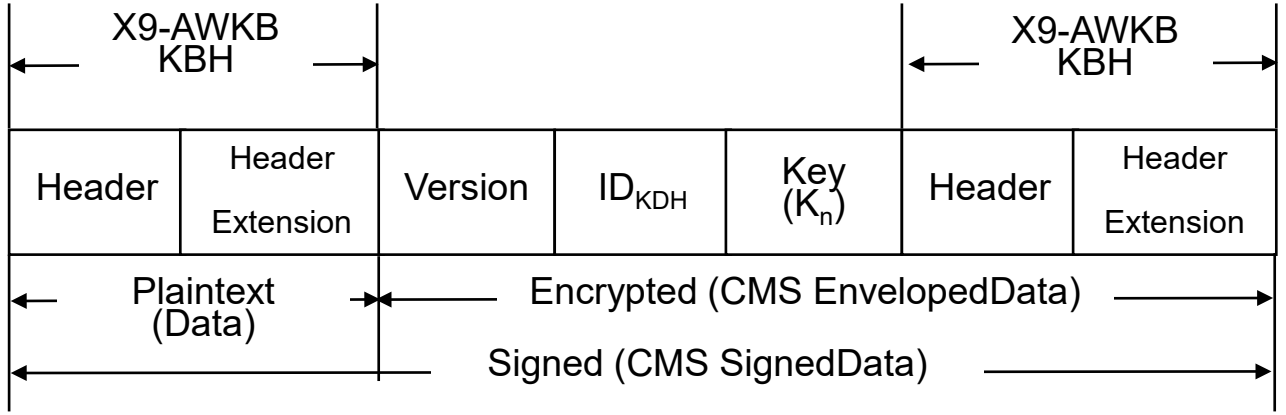
X9.143 / ISO 20038 updates planned

- **Multi-level key derivation control**
  - For complex systems
- **Post Quantum Cryptography (PQC)**
  - Carry private keys, public key certificates
- **More flexible length**
  - PQC keys are BIG
- **Adapt to changing key wrap views**
  - Variant deprecated in 2018
  - Add AESKW now



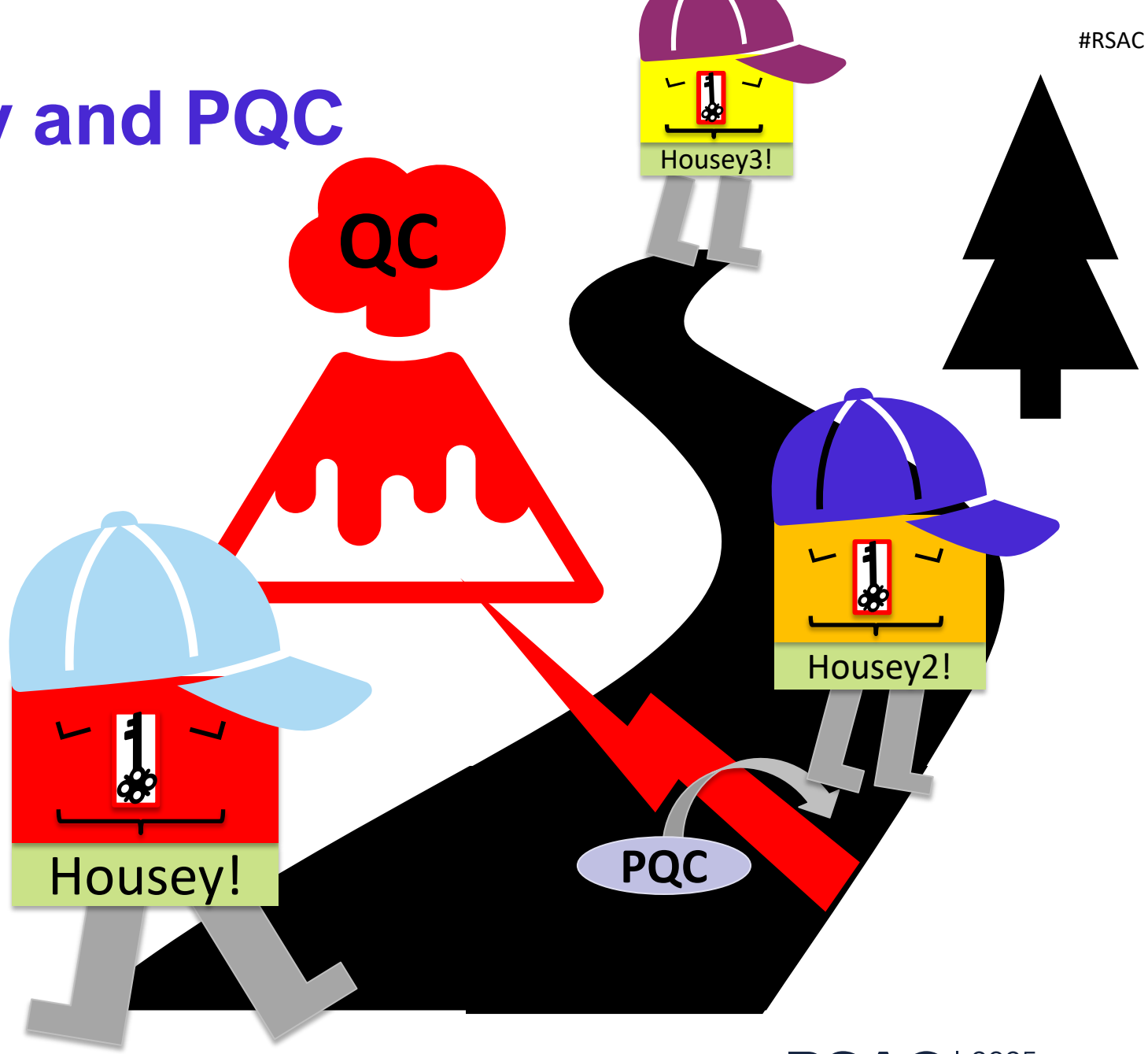
X9.139 (TR-34 evolution)

- **Encryption**
  - ECC support.
    Add ECDH key agreement
- **Authentication**
  - ECC support. Add EC-DSA
- **Elliptic Curves**
  - NIST P Curves and Brainpool P Curves
- **Compatibility**
  - Backward compatible with TR-34 supporting RSAES-OAEP and RSASSA-PKCS1-v1_5



Cryptographic agility and PQC

- **Build to last**
  - your ecosystem
  - your policy (attributes)
- **Be ready to change**
  - encryption
  - integrity protection



RSAC 2025 Conference

Wrapping it up

Many Voices.
One Community.

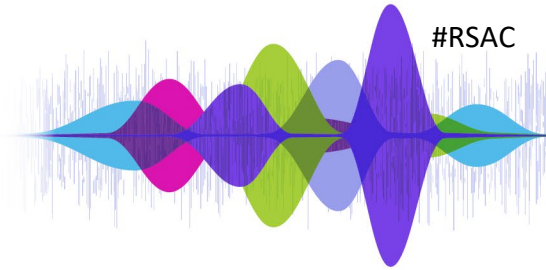
Apply what you have learned today

- Immediate actions
  - Check your certificates attributes, mechanisms, lifetimes
  - Check your key ‘container’ attributes and enforcement
    - Can you turn ‘security-optional’ containers into key blocks, or migrate to a key block solution?

Next 3 months

- Evaluate key exchange with partners for
  - attributes being tight enough
  - Implementations that enforce the attributes
- Evaluate TLS
  - Mechanisms list
  - Security of private keys
  - Endpoint access, proximity to application
- Cryptographic agility
  - consider implementations that allow policy-driven cryptography choices

Summary



- Key Blocks are the gold standard of key protection

Make sure your keys have Integrity, Proof (of Authentication), Encryption, Attributes, Secure Ecosystem (IPEAS).

- Key Protection is a Process not a Product

Build a trusted ecosystem of systems and controls.

- Action Items

- Evaluate Key protection
- Transition to secure key containers
- Design for cryptographic agility
- PQC readiness

RSAC™ 2025 Conference

Many Voices.
One Community.

Richard Kisley
Chief Engineer, HSM Development, IBM Corporation

Joachim Vance
Chief Security Architect, Distinguished Engineer, Veifone, Inc.

Thank you!

Richard Kisley: https://www.linkedin.com/in/kisley/

Joachim Vance: https://www.linkedin.com/in/joachimvance/